



PERSONAL EXPENSE TRACKER – DATA SCIENCE CAPSTONE PROJECT

ANALYZING AND PREDICTING PERSONAL SPENDING PATTERNS

EXECUTIVE SUMMARY

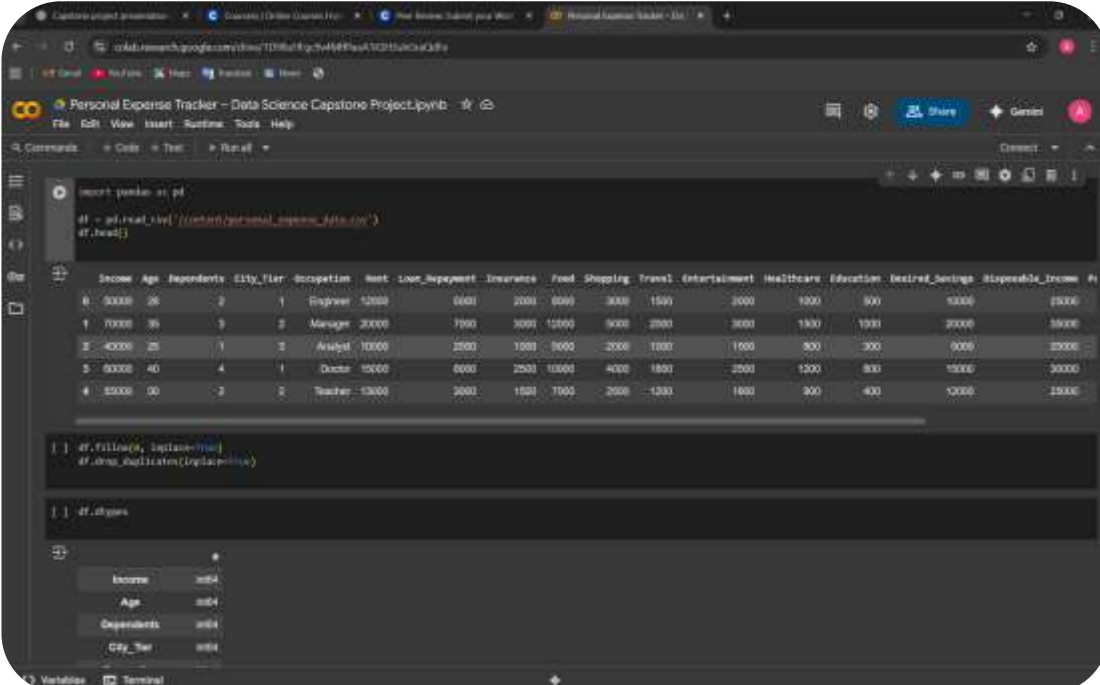
- Aim: Predict potential savings based on income, expenses, lifestyle.
- Dataset: Personal Expense Tracker with 19 columns.
- Methodology : Data Collection -> Data Cleaning -> EDA ->SQL Analysis -> Predictive Modeling -> Dashboard & Map
- Key Findings : Age , Income, City Tier affect savings ;Random Forest predicts well.

INTRODUCTION

- Problem: Individuals struggle to manage finances.
- Objective: Identify patterns & predict potential savings.
- Relevance: Helps budget planning.
- Placeholder: Optional infographic

DATA COLLECTION & WRANGLING

- Data Source :- CSV FILE
- Cleaning :- Handle Missing Values, drop duplicates , convert categorical
- Placeholder: Table Screenshot of dataset head.



The screenshot shows a Jupyter Notebook titled "Personal Expense Tracker - Data Science Capstone Project.ipynb". The code cell contains the following Python code:

```
import pandas as pd  
  
df = pd.read_csv('data/personal_expense_data.csv')  
df.head()
```

The output of the code is a table showing the first five rows of the dataset. The columns are: Income, Age, Dependents, City_Tier, Occupation, Rent, Loan_Repayment, Insurance, Food, Shopping, Travel, Entertainment, Healthcare, Education, Desired_Savings, and Disposable_Income.

	Income	Age	Dependents	City_Tier	Occupation	Rent	Loan_Repayment	Insurance	Food	Shopping	Travel	Entertainment	Healthcare	Education	Desired_Savings	Disposable_Income
0	50000	26	2	1	Engineer	12000	6000	2000	8000	3000	1500	2000	1000	500	10000	15000
1	70000	36	3	2	Manager	20000	7500	3000	12000	5000	2500	3000	1500	1000	20000	18000
2	40000	25	1	2	Analyst	10000	2500	1000	6000	2000	1800	1600	500	300	9000	23000
3	60000	40	4	1	Doctor	15000	8000	2500	10000	4000	1800	2500	1200	800	15000	30000
4	80000	30	2	2	Teacher	13000	3000	1500	7000	2000	1200	1800	500	400	12000	25000

Below the table, there are two empty code cells with the following code:

```
[ ] df.fillna(0, inplace=True)  
df.drop_duplicates(inplace=True)
```

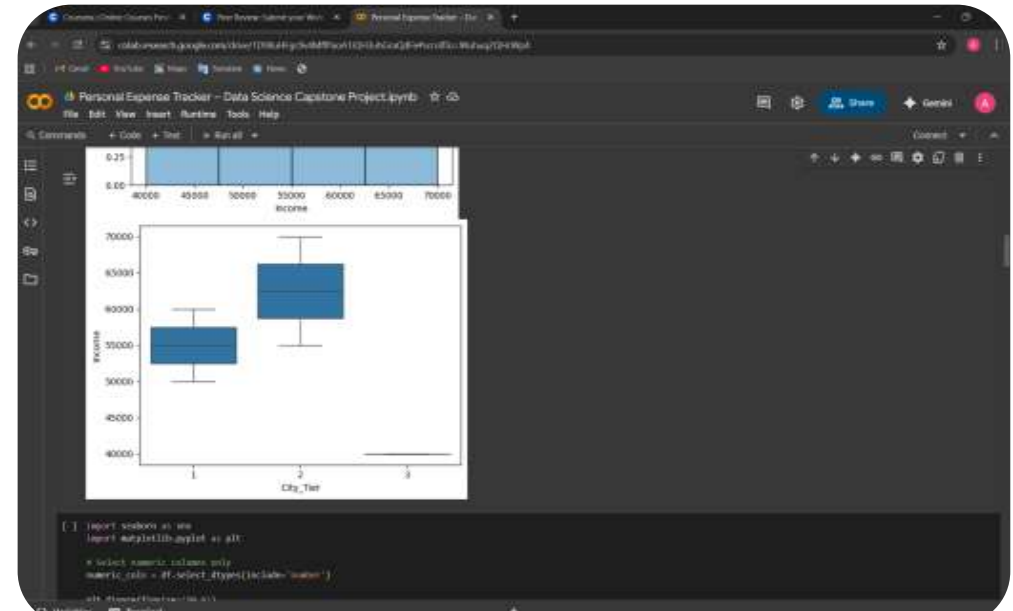
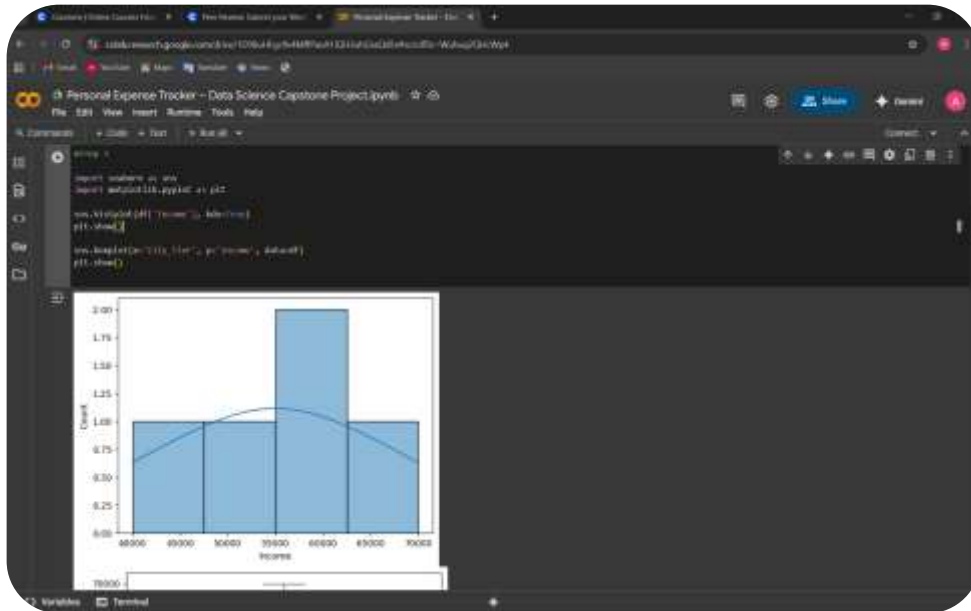
```
[ ] df.dtypes
```

The output of the code is a table showing the data types of the columns:

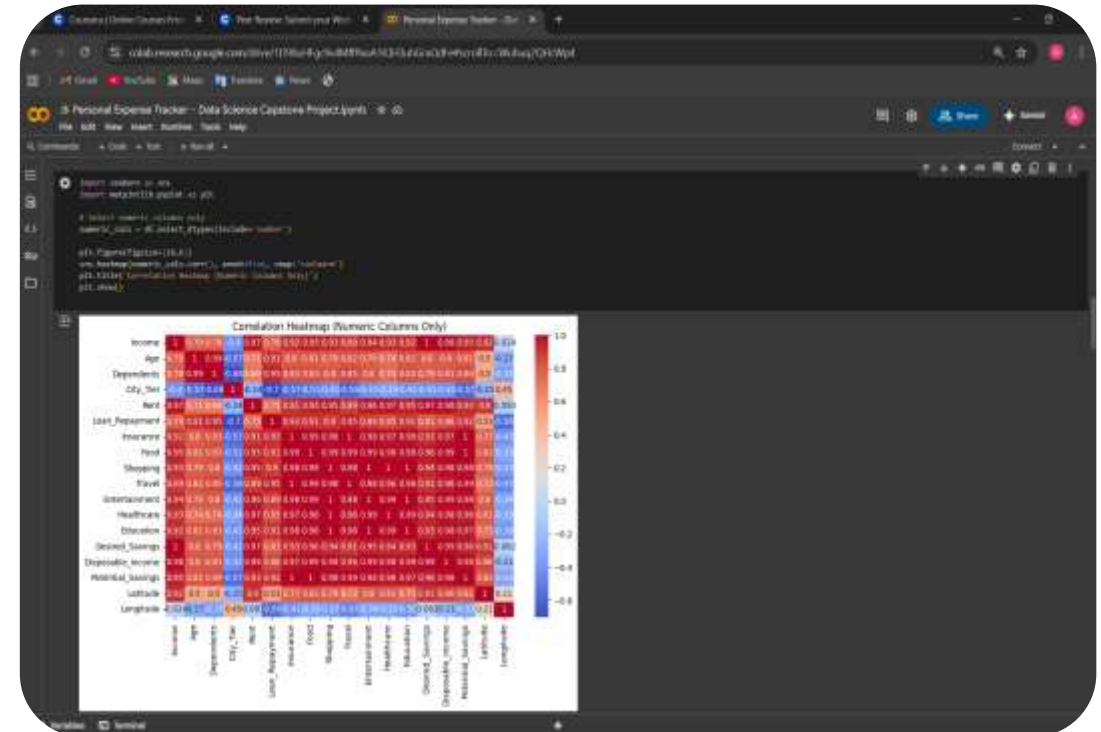
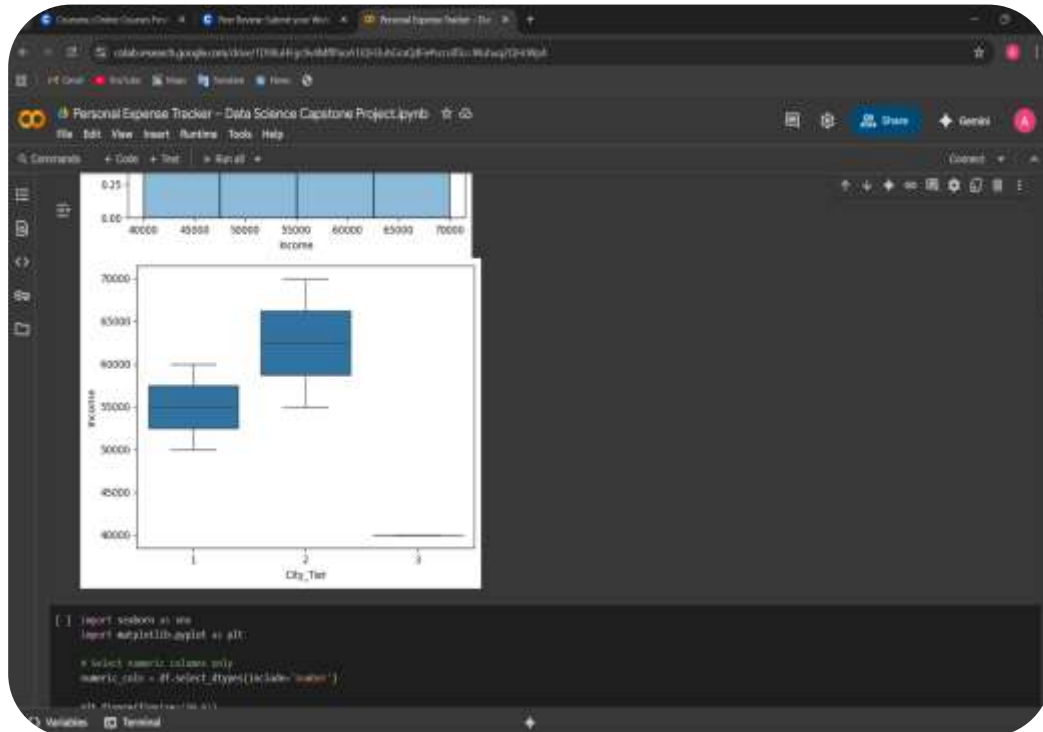
	Income	Age	Dependents	City_Tier
	int64	int64	int64	int64

EDA & INTERACTIVE VISUAL ANALYTICS

- Techniques: Histograms, Boxplots, Correlation Heatmaps, Pairplots.
- Insights: Income higher in City Tier I; Age correlates with savings; Rent & Loans affect Disposable Income.
- Placeholder: Include heatmap, histogram, boxplot images.

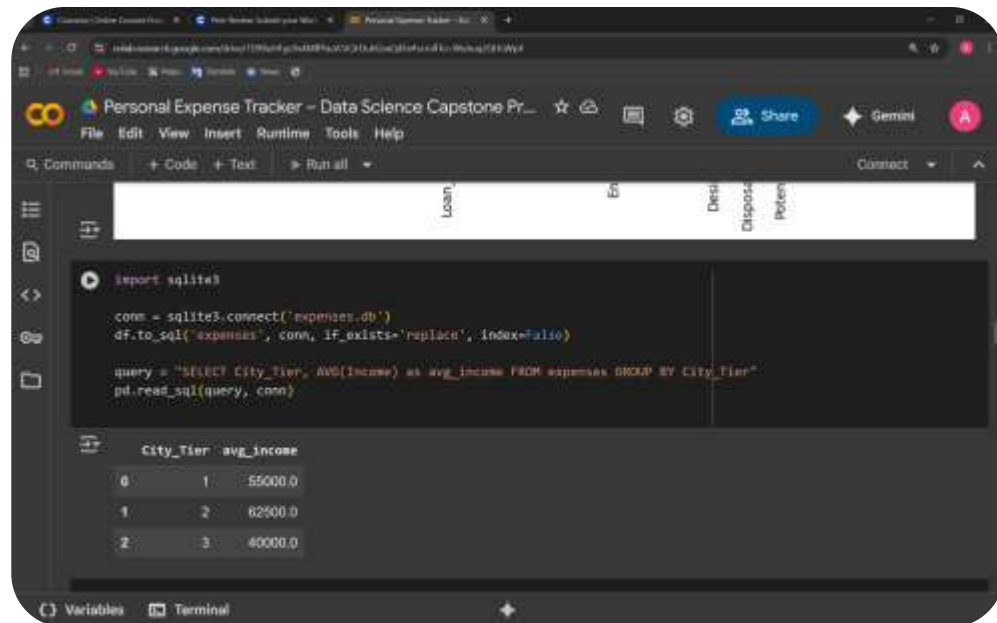


EDA & INTERACTIVE VISUAL ANALYTICS



PREDICTIVE ANALYSIS METHODOLOGY

- Target: Potential_Savings > 5000 (binary).
- Features: Income, Age, Rent, Dependents
- Model: Random Forest
- Train/Test: 80% / 20%
- Performance metric: Accuracy & Confusion Matrix



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes the Google Colab logo and the title 'Personal Expense Tracker - Data Science Capstone Project'. The notebook contains a code cell with the following SQL queries:

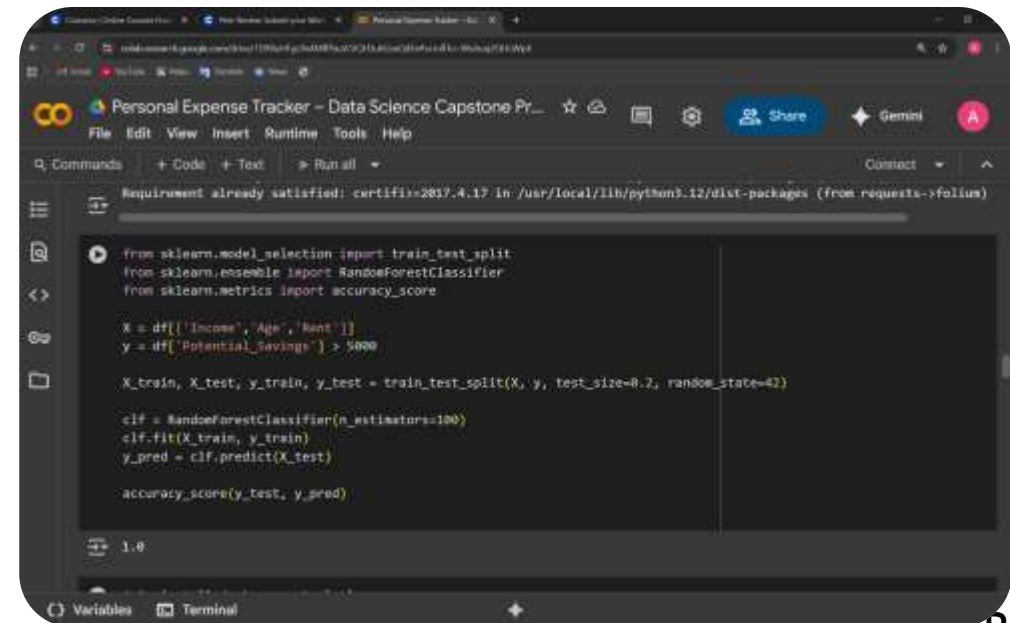
```
import sqlite3

conn = sqlite3.connect('expenses.db')
df.to_sql('expenses', conn, if_exists='replace', index=False)

query = "SELECT City_Tier, AVG(Income) as avg_income FROM expenses GROUP BY City_Tier"
pd.read_sql(query, conn)
```

Below the code cell, a table is displayed with the following data:

City_Tier	avg_income
0	1 55000.0
1	2 62500.0
2	3 40000.0



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes the Google Colab logo and the title 'Personal Expense Tracker - Data Science Capstone Project'. The notebook contains a code cell with the following Python code:

```
Requirement already satisfied: certifi==2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->folium)

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

X = df[['Income', 'Age', 'Rent']]
y = df['Potential_Savings'] > 5000

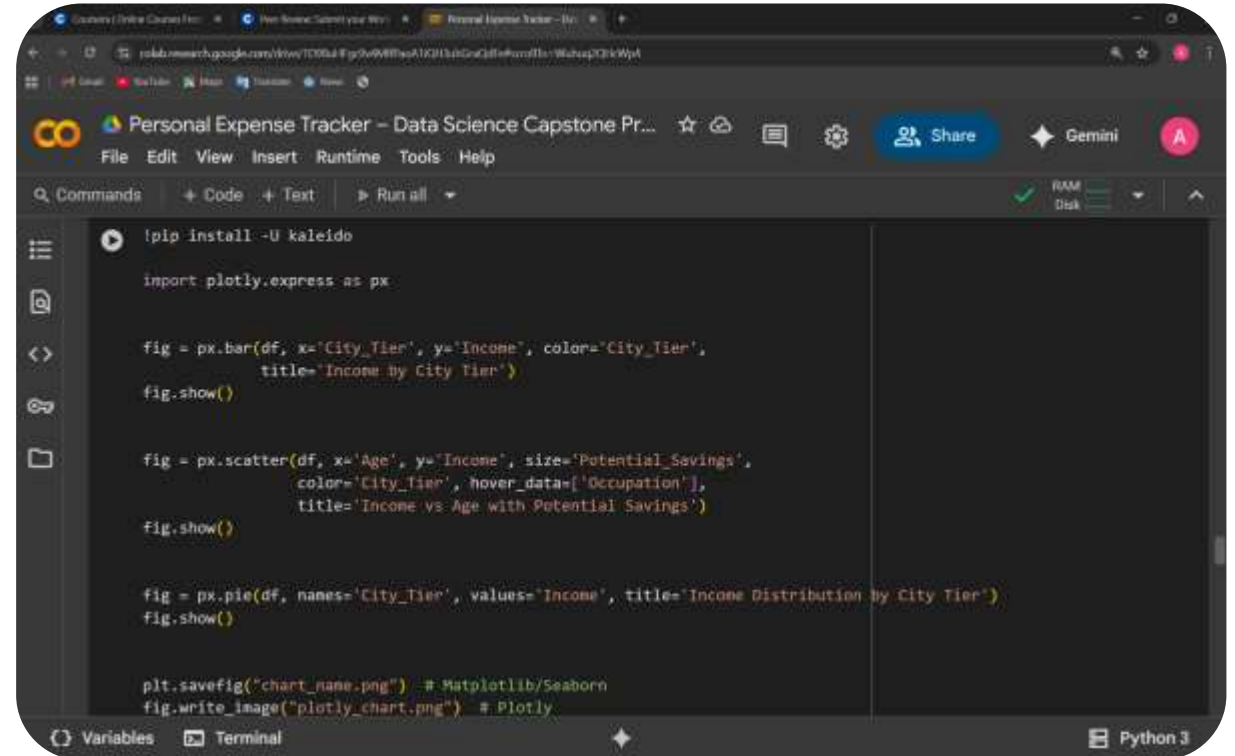
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf = RandomForestClassifier(n_estimators=100)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

accuracy_score(y_test, y_pred)
```

EDA WITH VISUALIZATION RESULTS

- Charts: Bar, Scatter, Pie, Pairplots
- Insights: Income distribution, spending patterns, correlations
- Placeholder: Chart screenshots



The screenshot shows a Jupyter Notebook titled "Personal Expense Tracker - Data Science Capstone Pr...". The code in the notebook is as follows:

```
!pip install -U kaleido

import plotly.express as px

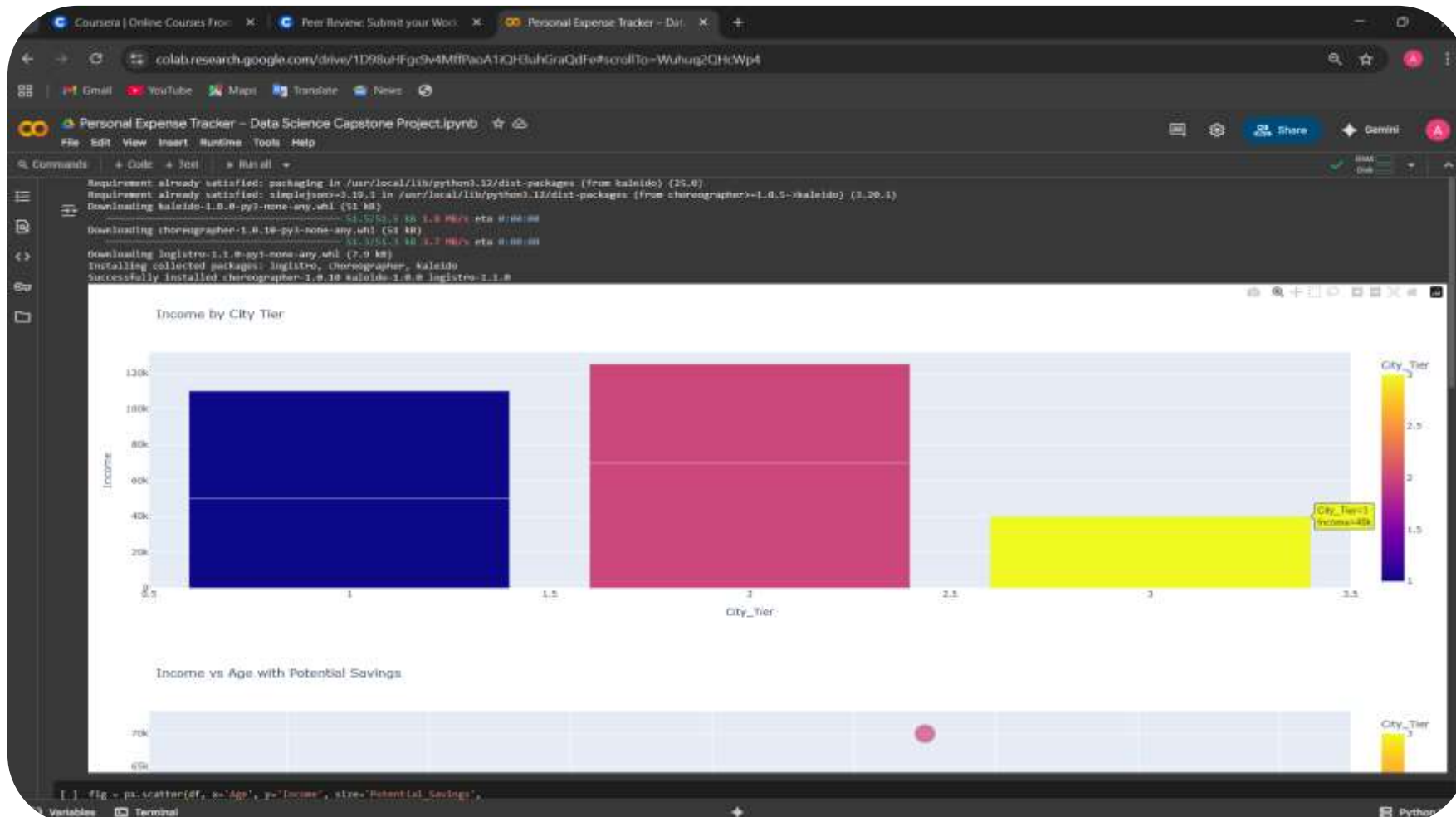
fig = px.bar(df, x='City_Tier', y='Income', color='City_Tier',
             title='Income by City Tier')
fig.show()

fig = px.scatter(df, x='Age', y='Income', size='Potential_Savings',
                 color='City_Tier', hover_data=['Occupation'],
                 title='Income vs Age with Potential Savings')
fig.show()

fig = px.pie(df, names='City_Tier', values='Income', title='Income Distribution by City Tier')
fig.show()

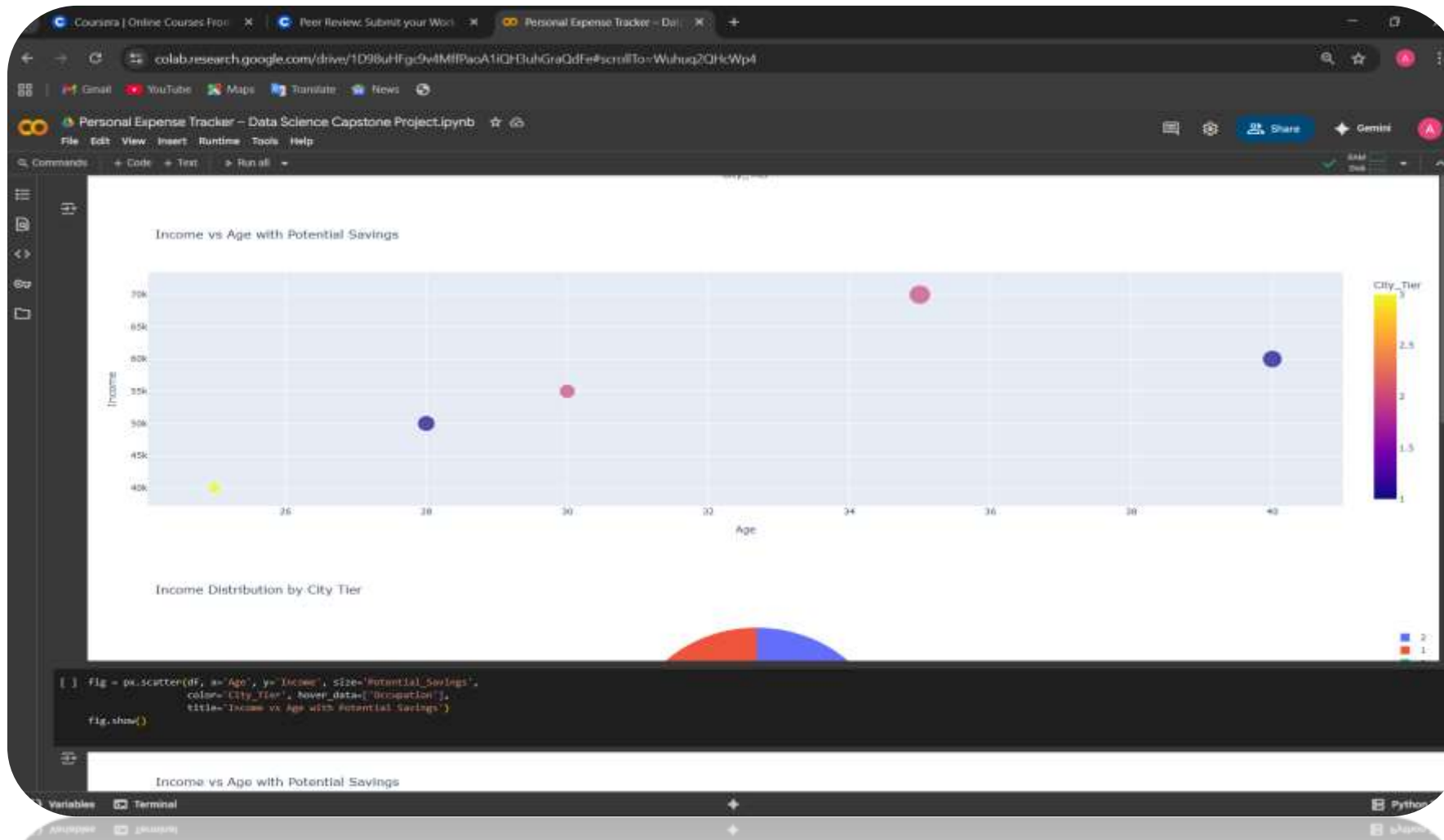
plt.savefig('chart_name.png') # Matplotlib/Seaborn
fig.write_image("plotly_chart.png") # Plotly
```


EDA WITH VISUALIZATION RESULTS



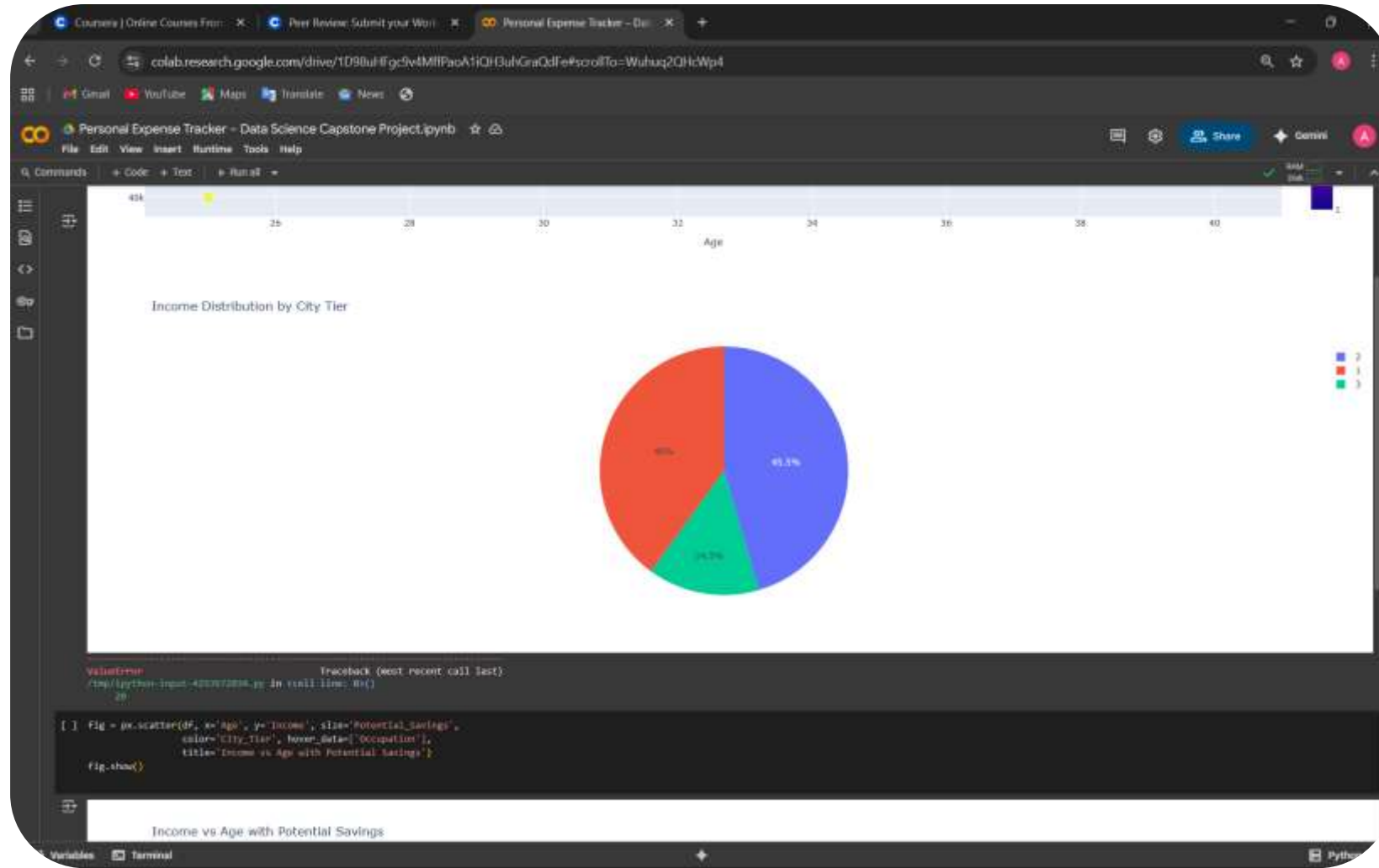
Presented By :Aanchal Gor

EDA WITH VISUALIZATION RESULTS



Presented By :Aanchal Gor

EDA WITH VISUALIZATION RESULTS



Presented By :Aanchal Gor

FUTURE WORK

Integration with SQL Databases

Automating data collection directly from SQL databases.

Creating optimized queries for faster retrieval and preprocessing.

Advanced Predictive Modeling

Applying machine learning models (Regression, Classification, Time Series).

Using historical SQL data to improve accuracy.

Scalability & Real-Time Predictions

Building real-time dashboards connected with SQL + ML pipelines.

Enabling live predictive insights for decision-making.

Data Visualization & BI Tools

Power BI/Tableau integration with SQL + predictive models.

Interactive visual reports for better understanding.

Continuous Model Improvement

Periodic retraining using updated SQL datasets.

Feedback loop for improving prediction performance

CONCLUSION

Key Findings :- Income, Age, Rent, City Tier influence savings; dashboards/maps help understanding

Future Work: Deploy web app, include more lifestyle variables

CREATIVITY & INNOVATIVE INSIGHTS

Extra Features: Interactive charts, Folium map, highlight patterns (e.g., students save less, doctors save more)

Future Work: Infographic of insights/trends



THANK YOU